



≡

HealthCare

曾得旭 4108029019

蔡怡欣 4108029018

許雅筑 4108029013

01

62



專案名稱：HealthCare



REUSE!

7個App資料夾



- Blog: 健身部落格
- Fit-forum: 健身影片
- Contact: 聯絡後臺管理員
- Discussion_forum : 論壇
- Calories: 卡路里攝取記錄
- Recipe: 分享健康食譜
- Profile : 用戶個人資料





- **WRITE POST** 寫帖文
- **COMMENT** 留言
- **LIKE** 按贊
- **SHOW MY POST** 顯示我的帖文





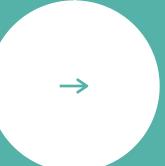
- **EXPERIENCE SHARING**
健身經驗分享
- **FITNESS/YOGA/HEALTHY DIET VIDEOS
SHARING BY 3 LEVELS**
影片分享，依據初級，中級，高級分類





CONTACT

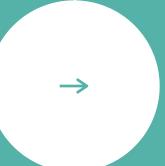
- SEND MESSAGES TO ADMIN
聯絡管理員 報告錯誤/反饋/建議





DISCUSSION_ FORUM

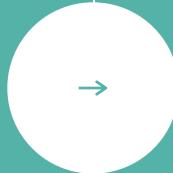
- ADD FORUM TOPICS
新增論壇話題
- COMMENT FORUM
論壇話題留言





CALORIES

- RECORD TOTAL CALORIES CONSUMED
記錄攝取的總卡路里
- RECORD CALORIE GOAL
記錄卡路里攝取目標
- RECORD FOOD CONSUMED TODAY
記錄當天攝取的食物
- ADD FOOD ITEM
新增食物類別
- SEARCH FOOD ITEM
搜尋查看食物類別
- SHOW ALL FOOD ITEM
顯示所有食物類別





RECIPE

- ADD RECIPE 新增健康食譜
- SHARE RECIPE 分享呈現健康食譜





PROFILE

- **USER INFO** 你的個人資料
- **MODIFY PROFILE** 更新你的個人資料
- **MODIFY CONTACT** 更新你的聯繫方式





技術介紹



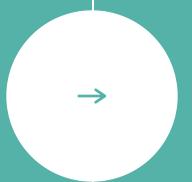
LOGIN, REGISTER, LOGOUT

獨立介紹





LOGIN



Login

Username...

Password...

Remember me

Login

Don't have an account? [Sign Up](#)

or login with:

- [GitHub](#)
- [Facebook](#)

LOGIN 自己寫的

1. 用Response + Cookie實現
2. 功能1小時後失效！

#使用Django Auth設計
#其他基本功能我們也有實現

用Response + Cookie實現

```
def LoginPage(request, page=''):
    if request.user.is_authenticated:
        if page == '':
            redirect('homepage')
        else:
            redirect(page)
    else:
        # if request.session.test_cookie_worked():
        #     request.session.delete_test_cookie()
        #     messages.info(request, "cookie supported")
        # else:
        #     messages.info(request, "cookie not supported")
        # request.session.set_test_cookie()
        if request.method == 'POST':
            username = request.POST.get('username')
            password = request.POST.get('password')
            user = authenticate(request, username=username, password=password)
            if user is not None:
                login(request, user)
                if request.POST.get('remember_me'):
                    if page == '':
                        response = redirect('homepage')
                    else:
                        response = redirect(page)

                    # expires = 'Thu, 28-May-2021 08:53:06 GMT' # 24小时 格林威治时间
                    # expires = datetime.datetime(2021, 5, 28, 23, 44, 55))
                    expires = 60 * 60 * 24
                    max_age = 60 * 60 * 24
                    response.set_cookie('c_username', username, expires=expires, max_age=max_age)
                    response.set_cookie('c_password', password, expires=expires, max_age=max_age)
                    return response
            else:
                if page == '':
                    return redirect('homepage')
                else:
                    return redirect(page)
            else:
                messages.warning(request, 'Username or password is incorrect')

    if request.COOKIES.get('c_username'):
        context = {
            'c_username': request.COOKIES['c_username'],
            'c_password': request.COOKIES['c_password'],
        }
    else:
        context = {}
    return render(request, 'login/login.html', context)
```

```
urlpatterns = [
    path('login/<str:page>', LoginPage),
    path('login/', LoginPage, name='login'),
    path('login/post_homepage', LoginPage, name='login/post_homepage'),
    path('login/blog_homepage', LoginPage, name='login/blog_homepage'),
    path('login/blog_show_my_posts', LoginPage, name='login/blog_show_my_posts'),
    path('login/blog_weight_lost', LoginPage, name='login/blog_weight_lost'),
    path('login/blog_sport', LoginPage, name='login/blog_sport'),
    path('login/home', LoginPage, name='login/home'),
    #####include('allauth.urls'))]
```

自己寫的路徑
登入後導入指定的畫面



Login

Username...

Password...

Remember me

Login

Account was created for
helen

Don't have an account? [Sign Up](#)

or login with:

- [GitHub](#)
- [Facebook](#)

Login

Username...

Password...

Remember me

Login

Username or password is
incorrect

註冊成功後，會跳轉至登
入畫面，並傳MESSAGE

錯誤會傳MESSAGE



Menu:

- [Sign In](#)
- [Sign Up](#)

登入

Please sign in with one of your existing third party accounts. Or, [sign up](#) for a `http://127.0.0.1:8000/` account and sign in below:

- [GitHub](#)
- [Facebook](#)

或

使用者名稱：

密碼：

記住我：

[忘記密碼了？](#) [登入](#)

1. 實現 GitHub 登入

2. 實現 Facebook 登入

3. 實現 Django Auth 登入

LOGIN 利用 ALLAUTH

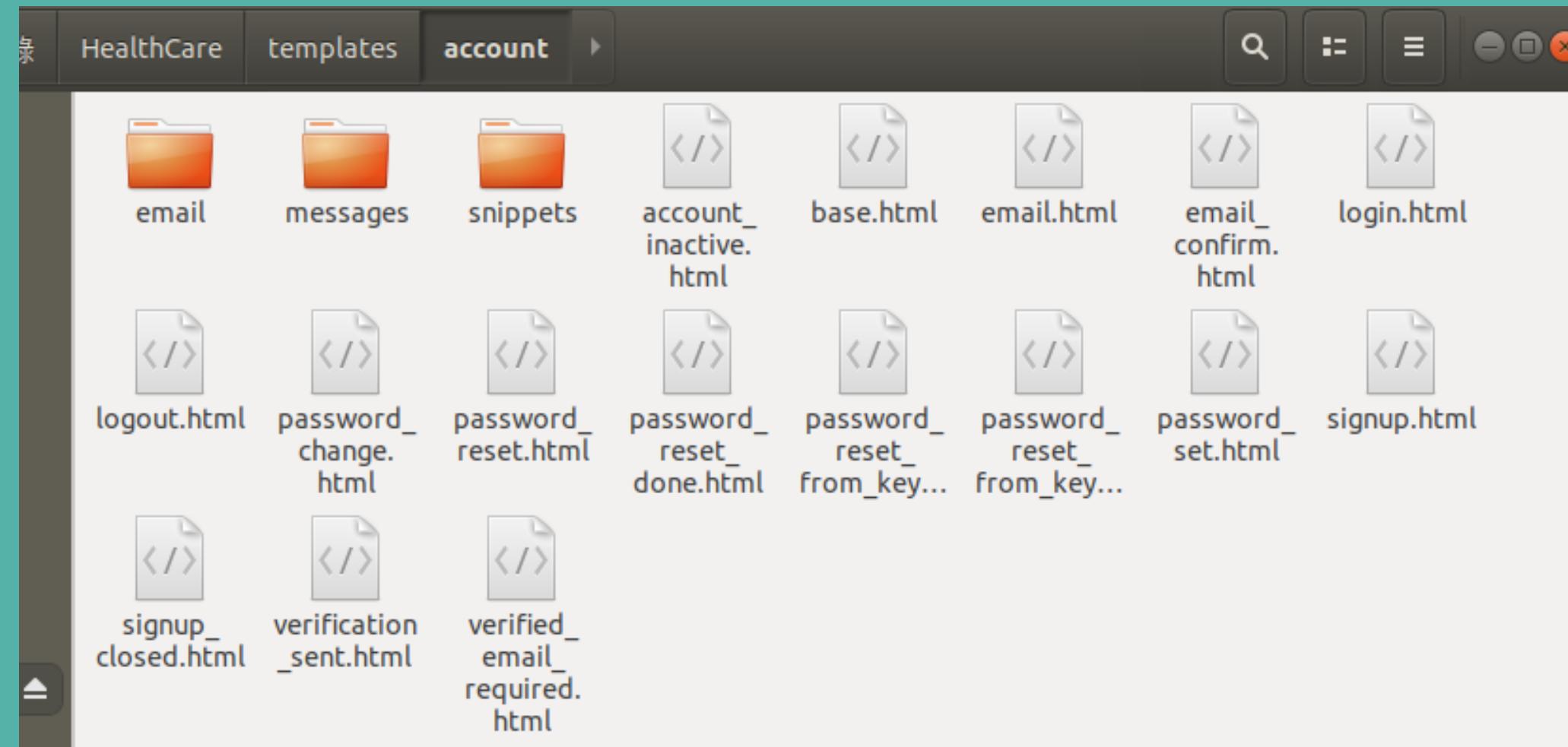
1. 實現GitHub登入

The screenshot shows the GitHub login page. At the top is the GitHub logo. Below it is a blue button with a white 'GH' icon. Text reads "Sign in to GitHub to continue to HealthCare". A large input field for "Username or email address" is empty. Below it is a password input field. To the right of the password field is a "Forgot password?" link. A green "Sign in" button is at the bottom. A link "New to GitHub? Create an account." is at the bottom left. At the very bottom are links for "Terms", "Privacy", "Security", and "Contact GitHub".

2. 實現Facebook登入

The screenshot shows the Facebook login page. At the top is the text "登入 Facebook". Below it is a red error box with the message "已封鎖不安全的登入嘗試" and "您無法從不安全的頁面存取或登入此應用程式。請嘗試以 https:// 重新載入網頁". Below the error box are fields for "電子郵件地址或手機號碼" and "密碼". A blue "登入" button is at the bottom. A "忘記密碼？" link is at the bottom right.

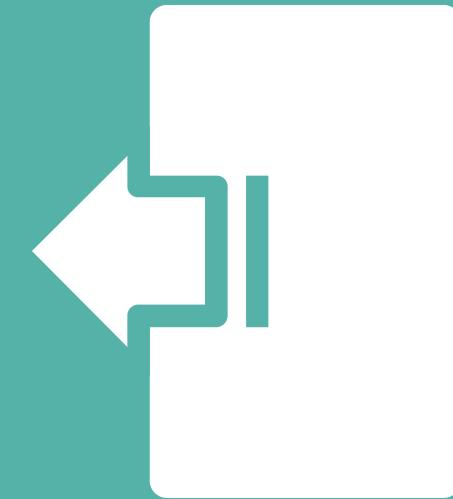
LOGIN 利用 ALLAUTH



ALLAUTH 其他畫面也有導入



REGISTER

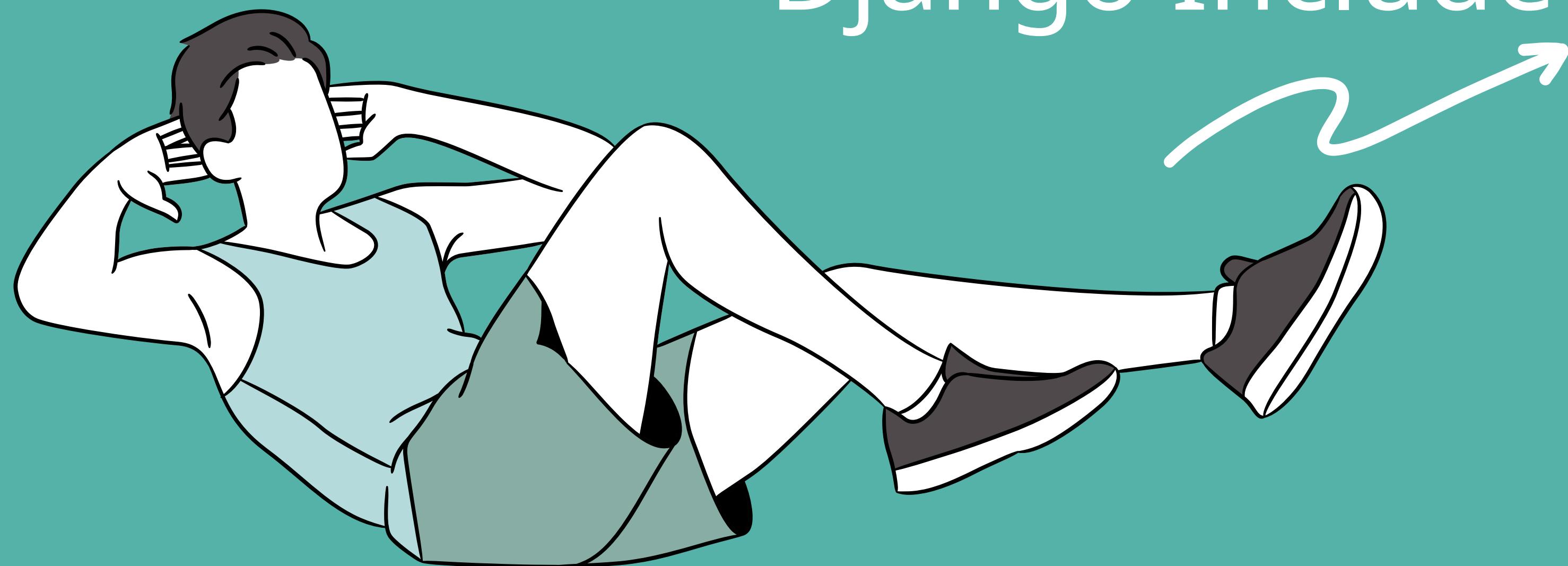


LOGOUT

與 LOGIN 同理



Django Include



```
path('calories/', include('calories.urls')),  
url(r'^myblog/', include('blog.urls')),  
url(r'^recipe/', include('recipe.urls')),  
url(r'^fitforum/', include('fitforum.urls')),  
url(r'^contact/', include('contact.urls')),  
url(r'^discuss/', include('Discussion_forum.urls')),  
url(r'^captcha/', include('captcha.urls')),
```

在app裡開url檔案，
通過include與HealthCare做聯動



Django Simple Captcha

*pip install django-simple-captcha





```
53         category2 = forms.ChoiceField(label='Message Subject 2', choices=choices, required=True)
54
55     class ContactForm3(ContactForm2):
56         name = forms.CharField(label='Your name', max_length=50)
57         email = forms.EmailField(label="Email")
58         message = forms.CharField(label='Message', widget=forms.Textarea)
59         captcha = CaptchaField()
60
61     form = ContactForm3
62
```



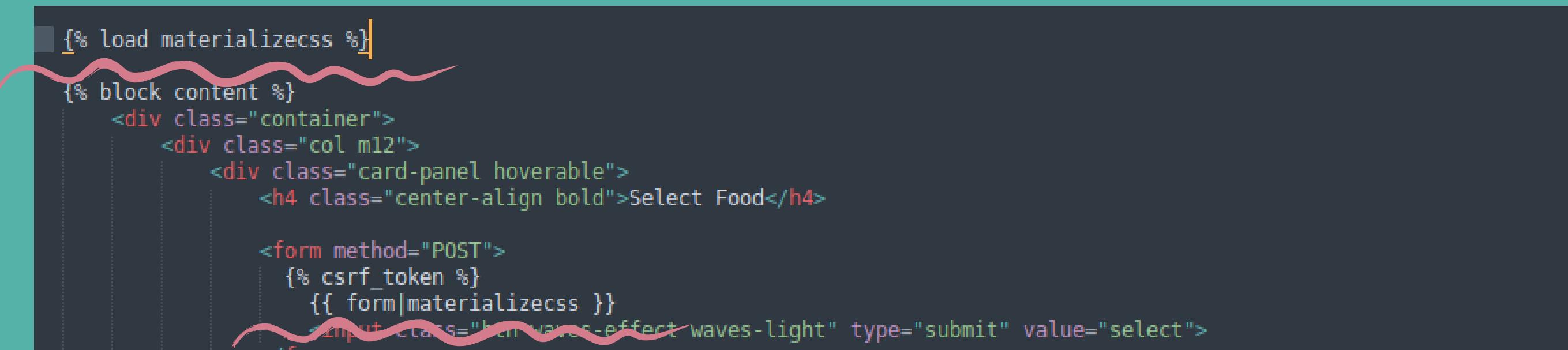
用CaptchaField做機器人驗證

Materializecss For Web Design

*`pip install django-materializecss`



#我們用Marerialize CSS 來設計我們的網頁



```
{% load materializecss %}

{% block content %}
    <div class="container">
        <div class="col m12">
            <div class="card-panel hoverable">
                <h4 class="center-align bold">Select Food</h4>

                <form method="POST">
                    {{ csrf_token }}
                    {{ form|materializecss }}
                    <input class="btn waves-effect waves-light" type="submit" value="select">
                </form>
            </div>
        </div>
    </div>
</block>
```



這個套件可以直接幫我們套
Materializecss的模板

HTML手機畫面適配



≡

```
</nav>
<ul class="side-nav" id="nav-mobile">
    {% with url_name=request.resolver_match.url_name %}
        <li {% if url_name == 'blog_homepage' %}class="active"{% endif %}><a href="{% url 'blog_homepage' %}"><i class="material-icons">restaurant_menu</i>Nutrition</a></li>
        <li {% if url_name == 'blog_sport' %}class="active"{% endif %}><a href="{% url 'blog_sport' %}"><i class="material-icons">watch</i>Sport</a></li>
        <li {% if url_name == 'blog_weight_lost' %}class="active"{% endif %}><a href="{% url 'blog_weight_lost' %}"><i class="material-icons">directions_bike</i>Weight Lost</a></li>
```





*`HttpResponseRedirect(request.META.get('HTTP_REFERER'))`

Request and response objects



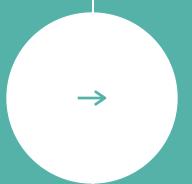
```
def LogOutPage(request):  
    logout(request)  
    return HttpResponseRedirect(request.META.get('HTTP_REFERER'))
```



導回前一頁



PhoneNumberField 套件



```
class Contact(models.Model):  
    user = models.ForeignKey(User, on_delete=models.CASCADE)  
    phone_number = PhoneNumberField(blank=True)  
    email = models.CharField(max_length=100, null=True, blank=True)  
    website = models.URLField(null=True, blank=True)
```



檢查使用者輸入的電話號碼是否符合格式



Database Using MySQL



活动 窗口 终端 ▾

週六 00 : 39 •

HealthCare - settings.py [HealthCare]

zh ▾ 🔍 ⚡ ⚡ ⚡

File Edit View Navigate Code Git Help

Project addInDiscussion.html × settings.py ×

Commit

Pull Requests

Database changed

mysql> show tables;

+-----+ <td>Tables_in_HealthCare_db</td>	Tables_in_HealthCare_db
	Discussion_forum_discussion
	Discussion_forum_forum
	account_emailaddress
	account_emailconfirmation
	auth_group
	auth_group_permissions
	auth_permission
	auth_user
	auth_user_groups
	auth_user_user_permissions
	blog_categories
	blog_comment
	blog_likes
	blog_post
	calories_food
	calories_postfood
	calories_profile
	captcha_captchastore
	contact_contact
	contact_message_subject_1
	contact_message_subject_2
	django_admin_log
	django_content_type
	django_migrations
	django_session
	django_site
	fitforum_level
	fitforum_post
	recipe_food

107:20 LF UTF-8 4 spaces Python 3.6 final_v1 🔍 ⚡ ⚡ ⚡ Event Log

```
'ENGINE': 'django.db.backends.sqlite3',
'NAME': BASE_DIR / 'db.sqlite3',
},
'HealthCare_db': {
    'ENGINE': 'django.db.backends.mysql',
    'NAME': 'HealthCare_db',
    'USER': 'user',
    # 'PASSWORD': '',
    'HOST': 'localhost',
    'PORT': '',
    'OPTIONS': {
        'init_command': "SET sql_mode='STRICT_TRANS_TABLES'",
    },
},
},
# Auth Login
ACCOUNT_ACTIVATION_DAYS = 7
'HealthCare_db'
```

使用url參數傳遞及url反解



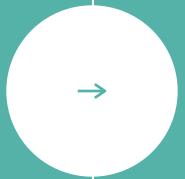


```
<div class="row mb-2 posts">
    {% for post in posts %}
        <div class="col-md-6">
            <div class="cards">
                <div class="row no-gutters border rounded flex-md-row mb-4 shadow-sm h-md-250">
                    <div class="col p-4 d-flex flex-column position-static">
                        <h3 class="my-1"><a href="{% url 'detail' post.slug %}">{{post.title}}</a>
                        </h3>
                        <div style="display:flex">
                            {% for tag in post.tags.all %}
                                <a href="{% url 'tagged' tag.slug %}" class="mr-1 badge badge-info">
                                    #{{ tag }}</a>
                            {% endfor %}
                        </div>
                        <p class="mb-auto">{{post.description}}</p>
                        <p class="mb-auto text-muted">{{post.published}}</p>
                    </div>
                </div>
            </div>
        </div>
    </div>
```



#方便設計網頁路徑

Model Field: Foreign Key





```
class Contact(models.Model):
    user_name = models.CharField(max_length=50, default="(No Name)")
    # message_subject = models.ForeignKey(message_subject_1, null=False, on_delete=models.SET_DEFAULT, default=4)
    message_subject2 = models.ForeignKey(message_subject_2, null=True, on_delete=models.SET_DEFAULT, default=6)
    user_email = models.CharField(max_length=50)
    user_message = models.CharField(max_length=100)
    status = models.BooleanField(default=False)
```

```
class Food(models.Model):
    name = models.CharField(max_length=200, null=False)
    quantity = models.PositiveIntegerField(null=False, default=0)
    calorie = models.FloatField(null=False, default=0)
    person_of = models.ForeignKey(User, null=True, on_delete=models.CASCADE)

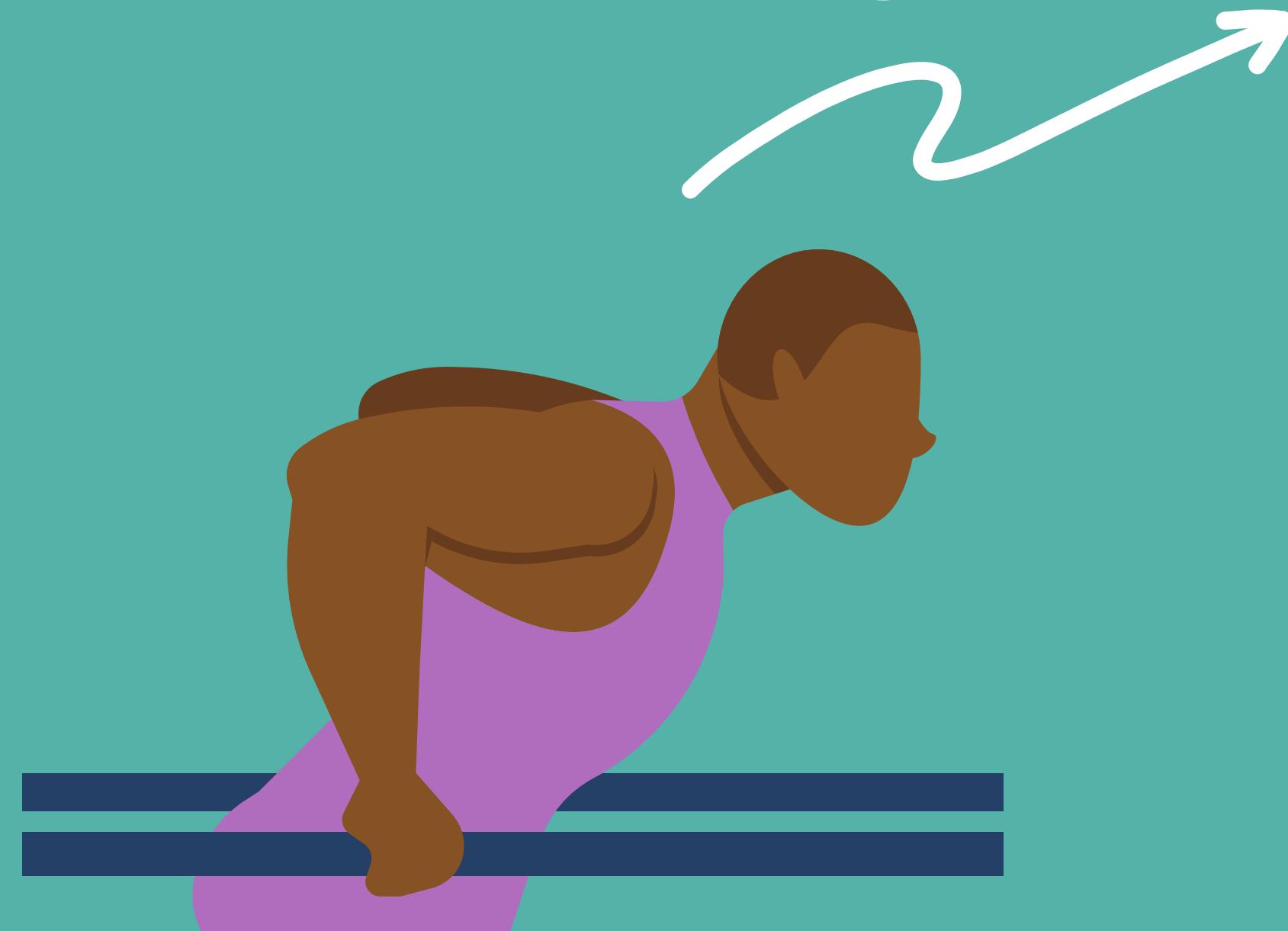
    def __str__(self):
```

```
17 class message_subject_2(models.Model):
18     category = models.ForeignKey(message_subject_1, null=True, on_delete=models.SET_NULL)
19     title = models.CharField(max_length=250, blank=False)
20     description = models.CharField(max_length=250, blank=True, null=True)
21
```

#根據使用情況設計不同的on_delete條件



Django forms: Image Field



≡

```
content = models.TextField(default='(empty)')
published = models.DateTimeField(auto_now_add=True)
pic = models.ImageField(upload_to='pics', null=True, blank=True, default='default.png')
```

```
152
153     MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
154     MEDIA_URL = '/media/'
155
```

```
47
48     ] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
49
```



#使用者可以上傳資料

*pip install django-taggit

Django Taggit



The screenshot shows a web application interface with a sidebar menu on the left and a central configuration panel.

左侧菜单 (Sidebar Menu):

- BLOG
 - Categoriss [新增](#)
 - Comments [新增](#)
 - Likess [新增](#)
 - Posts [新增](#)
- CALORIES
 - Foods [新增](#)
 - Post foods [新增](#)
 - Profiles [新增](#)
- CONTACT
 - Contacts [新增](#)
 - Message_subject_1s [新增](#)
 - Message_subject_2s [新增](#)
- DISCUSSION_FORUM
 - Discussions [新增](#)
 - Forums [新增](#)
- FITFORUM
 - Levels [新增](#)
 - Posts [新增](#)

中央面板 (Central Panel):

選擇 tag 來變更

動作: ----- 去 4 中

<input type="checkbox"/>	名稱
<input type="checkbox"/>	a
<input type="checkbox"/>	b
<input type="checkbox"/>	c
<input type="checkbox"/>	d

4 tags

右侧面板 (Right Panel):

變更 tag

a

名稱:

Slug:

TAGGED ITEMS

Tagged item: Grapefruit_test tagged with a

內容類型:

Object ID:

Tagged item: admin write post tagged with a

內容類型:

Object ID:

Tagged item: #3

內容類型:

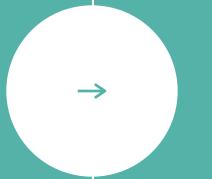
Object ID:

Tagged item: #4

內容類型:

#特色：根據Tag來分類Post

Django Forms



```
28
29
30 def create_form(catg1=None, catg2=None):
31
32     class ContactForm1(forms.Form):
33         choices = []
34         if catg1 is None:
35             message_sub1 = models.message_subject_1.objects.all()
36             for m in message_sub1:
37                 choices.append([str(m), str(m)])
38         else:
39             message_sub1 = models.message_subject_1.objects.get(title=catg1)
40             m = message_sub1
41             choices.append([str(m), str(m)])
42
43         category = forms.ChoiceField(label='Message Subject', choices=choices, required=True)
44
45     form = ContactForm1
46
47     if catg2 is not None:
48         class ContactForm2(ContactForm1):
49             message_sub2 = models.message_subject_2.objects.filter(category=catg2)
50             choices = []
51             for m in message_sub2:
52                 choices.append([str(m), str(m)])
53             category2 = forms.ChoiceField(label='Message Subject 2', choices=choices, required=True)
54
55         class ContactForm3(ContactForm2):
56             name = forms.CharField(label='Your name', max_length=50)
57             email = forms.EmailField(label="Email")
58             message = forms.CharField(label='Message', widget=forms.Textarea)
59             captcha = CaptchaField()
60
61         form = ContactForm3
62
63     return form
64
```

用了繼承的特性，根據使用者選的內容，組合並投放在不同的表單，可重複使用



#在template裡直接用
form.as_p 呼叫即可

Django Model Forms





```
class PostForm(forms.ModelForm):
    class Meta:
        model = Post
        fields = [
            'category',
            'title',
            'description',
            'content',
            'pic',
            'tags',
        ]
    # def __init__(self, *args, **kwargs):
    #     super(PostForm, self).__init__(*args, **kwargs)
    #     self.fields['title'].label='title'
    #     self.fields['description'].label ='description'
    #     self.fields['content'].label ='content'
    #     self.fields['pic'].label ='pic'
    #     self.fields['tags'].label ='tags'

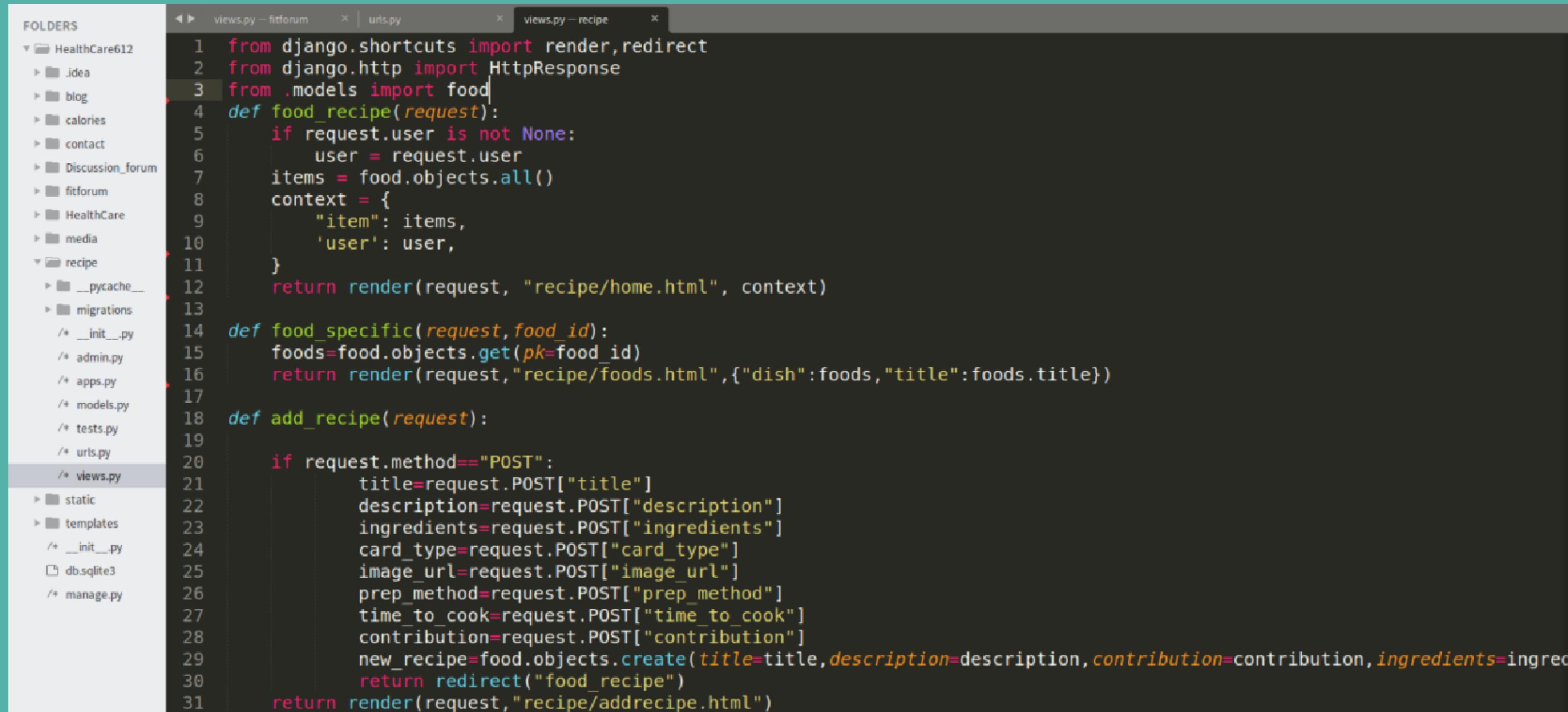

```

```
class CommentForm(forms.ModelForm):
    class Meta:
        model = Comment
        fields = [
            'name',
            'email',
            'body',
        ]
```



在HTML寫Form，然
後通過Views實現功能





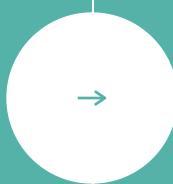
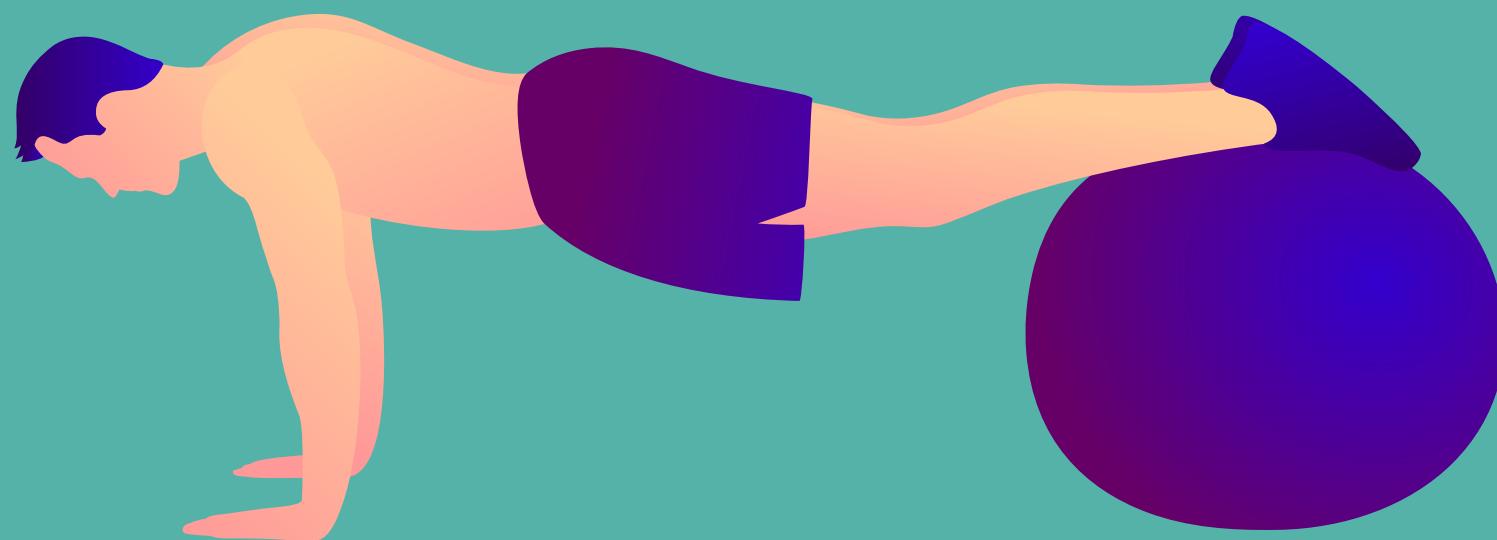
The screenshot shows the PyCharm IDE interface with the following details:

- Folders:** HealthCare612, .idea, blog, calories, contact, Discussion_forum, fitforum, HealthCare, media, recipe, __pycache__, migrations, __init__.py, admin.py, apps.py, models.py, tests.py, urls.py, views.py.
- Views.py - fitforum:** This tab contains the code for the fitforum application's views.
- Urls.py - recipe:** This tab contains the code for the recipe application's urls.
- Code Content:**

```
1 from django.shortcuts import render, redirect
2 from django.http import HttpResponseRedirect
3 from .models import food
4 def food_recipe(request):
5     if request.user is not None:
6         user = request.user
7         items = food.objects.all()
8         context = {
9             "item": items,
10            'user': user,
11        }
12     return render(request, "recipe/home.html", context)
13
14 def food_specific(request, food_id):
15     foods=food.objects.get(pk=food_id)
16     return render(request,"recipe/foods.html", {"dish":foods,"title":foods.title})
17
18 def add_recipe(request):
19
20     if request.method=="POST":
21         title=request.POST["title"]
22         description=request.POST["description"]
23         ingredients=request.POST["ingredients"]
24         card_type=request.POST["card_type"]
25         image_url=request.POST["image_url"]
26         prep_method=request.POST["prep_method"]
27         time_to_cook=request.POST["time_to_cook"]
28         contribution=request.POST["contribution"]
29         new_recipe=food.objects.create(title=title,description=description,contribution=contribution,ingredients=ingredients)
30         return redirect("food_recipe")
31     return render(request,"recipe/addrecipe.html")
```

#用View實現Form的功能

- get_object_or_404
- 内建Filter





```
55
56     @login_required(login_url='login/post_homepage')
57     def tagged(request, slug):
58         user = request.user
59         tag = get_object_or_404(Tag, slug=slug)
60         common_tags = Post.tags.most_common()[:4]
61         posts = Post.objects.filter(enabled=True).filter(tags=tag)
62         context = {
63             'tag': tag,
64             'common_tags': common_tags,
65             'posts': posts,
66             'user': user,
67         }
68
69         return render(request, 'blog/home.html', context)
70
71
72     def homepage(request):
73         if request.user is not None:
74             user = request.user
75             username = user.username
76             posts = Post.objects.all().filter(category='3').filter(enabled=True).order_by('-published', )
77             common_tags = Tag.objects.all()
78             form_c = CommentForm()
79             |
```



Django Filter





```
1 import django_filters
2 from django_filters import CharFilter
3 from .models import *
4
5 class FoodFilter(django_filters.FilterSet):
6     food_name = CharFilter(field_name = 'name' , lookup_expr = 'icontains',label='search food items')
7     class Meta:
8         model = Food
9         fields = ['food_name']
10
```



#用在搜尋功能，可以根據輸入的字眼進行搜尋

Django Signals



```
1  from django.db.models.signals import post_save
2  from django.contrib.auth.models import User
3  from .models import Profile, PostFood
4  def create_profile(sender, instance, created, **kwargs):
5      if created:
6          Profile.objects.create(person_of=instance)
7          print("profile created")
8
9  post_save.connect(create_profile, sender=User)
10
```

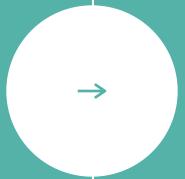
→

Django 提供了一組內置信號，讓 Django 本身通知用戶代碼某些操作，**django.db.models.signals.post_save** 在調用 Model 的 **save()** 方法之前或之後發送

當新增用戶後，會新增 Profile



Django Datetime





```
from .forms import SelectFoodForm, AddFoodForm, CreateUserForm, ProfileForm
from calories.models import *
from datetime import timedelta
from django.utils import timezone
from datetime import date
from datetime import datetime
from .filters import FoodFilter
```

```
21
22     #creating one profile each day
23     if date.today() > calories.date:
24         profile=Profile.objects.create(person_of=request.user)
25         profile.save()
26
27     calories = Profile.objects.filter(person_of=request.user).last()
28
```



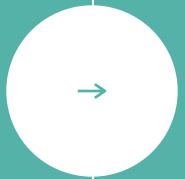
```
#querying all records for the last seven days
some_day_last_week = timezone.now().date() -timedelta(days=7)
records=Profile.objects.filter(date__gt=some_day_last_week,date__lt=timezone.now().date(),person_of=request.user)

context = {'form':form,'food_items':food_items,'records':records}
return render(request, 'calories/profile.html',context)
```

#取到用戶 7天內的資料



Django Tags



CUT

```
<div class="col l4 m6 s12 gallery-item gallery-expand gallery-filter {{ post.tags.all |cut:'<QuerySet  
[ ' | cut:'<Tag: ' | cut:'>' | cut:', ' | cut:']' }} ">  
    <div class="gallery-item__inner">
```

TRUNCATECHARS

```
<br>  
<span>{{ post.description | truncatechars:100 }}</span>  
<br>  
<span>{{ post.likes }} Likes</span>  
<!-- ##### -->
```

WITH

```
<ul class="right hide-on-med-and-down">  
    {% with url_name=request.resolver_match.url_name %}  
        <li {% if url_name == 'blog_homepage' %}class="active"{% endif %}><a href="{% url  
            'blog_homepage' %}">Nutrition</a></li>  
        <li {% if url_name == 'blog_sport' %}class="active"{% endif %}><a href="{% url 'blog_sport'  
            %}">Sport</a></li>  
        <li {% if url_name == 'blog_weight_lost' %}class="active"{% endif %}><a href="{% url  
            'blog_weight_lost' %}">Weight Lost</a></li>  
        <li><a class='dropdown-button' href="#" data-activates='feature-dropdown' data-belowOrigin="true"  
            data-constrainWidth="false">
```

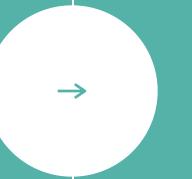
SAFE

```
66          {{ post.content | linebreaks | safe }}  
67      </h6>
```





IF, FOR, ELSE, URL, INCLUDE,
EXTENDS, BLOCK, FORLOOP,
COUNTER.....





Happy exercising!

